Лабораторна робота № 5

Тема роботи: . Програмування у середовищі MatLAB Script – файлів,

**Тема роботи:** Використання Script- файлів та діалогового режиму при програмування у середовищі MatLAB.

**Мета роботи**: Вивчення особливостей запису Script – файлів та операторів діалогового режиму при програмування у середовищі MatLAB.

Постановка задачі: Створити програму обчислень з використанням Script– файлу та діалогового режиму у середовищі MatLAB.

Вступ

1. *Створення Script - файлів*

1.1. Головні особливості Script- файлів

Як вже зазначалося, головні особливості Script- файлів e такими:

- Script-файли е незалежно (самостійно) виконуваними блоками onepaтopiв i команд;

- yci змінні, що використовуються в них, утворюють так званий робочий npocтip, який е загальним для ycix виконуваних Script- файлів; з цього випливає, що при виконанні кількох Script- файлів імена змінних у них повинні бути узгоджені, бо одне ім'я позначатиме у кожному з них той самий об’єкт обчислення;

- в них нема заголовка, тобто першого рядка певного вигляду i призначення;

- звернення до них не потребує вказання ніяких імен змінних: yci змінні формуються в результат виконання програми або сформовані paнiшe i icнyють у робочому просторі.

Слід зауважити, що робочий npocтip Script- файлів не є досяжним для файл-функцій, які використовуються в ньому. У файл-функціях неможливо використовувати значення, що надані змінним у Script-файлі, минаючи заголовок файл-функцій (через те, що yci змінні файл-функцій є локальними). Єдиною можливістю зробити так, щоб усередині файл-функції деяка змінна робочого простору зберігала своє значення i ім'я, є спеціальне оголошення цієї змінної у Script-файлі як *глобальної* за допомогою службового слова global. Kpiм того, аналогічний запис повинен міститися i у тексті М-файлу тієї файл-функції, яка буде використовувати значення відповідної змінної Script-файлу.

Наприклад, можна перебудувати файл-функцію обчислення функції у=f (х)= а\* sin(x)\* що була записана як:

function у = Fl(x,а)

% Процедура, що обчислює значення функції

% у = а\*sin(x)\*х^(2/3)

% Звернення у = Fl(x,а).

у = а\*sin(x) . \*х^(2/3);

вводячи коефіцієнт *а* як глoбaльну змінну:

function у = Fl(x)

% Обчислення функції

% у = а\*sin(x)\*х^(2/3)

%

global а

у = а\*sin(x) . \*х^(2/3);

Щоб використати нову файл-функцію Fl у Script-файлу, в останньому до звернення до цiєї функції повинна бути визначена стала *а* (задано її значення), а після цього записано рядок global а.

Якщо у одному рядку оголошуються кілька змінних як глобальні, вони повинні відділятися пробілами (не комами).

1.2. *Введення i виведення інформації у діалоговому режимі*

Для забезпечення взаємодії з користувачем протягом виконання М-файлу у системі MatLAB призначено такі команди:

*disp, sprintf, input, menu, keyboard, pause*.

Команда *disp* здійснює виведення значень вказаної змінної або вказаного тексту до командного вікна. Звернення до неї має вигляд:

*disp* (<змінна або текст в апострофах>).

Особливістю цiєї команди є те, що аргумент у неї може бути тільки один. Тому неможливо без спеціальних заходів здійснити виведення кількох змінних i, особливо, деяке сполучення тексту зi значеннями деяких змінних, що часто є необхідним для зручного подання інформації. Для усунення цього недоліку використовують кілька засобів.

Щоб вивести значення кількох змінних у один рядок (це є необхідним при утворенні таблиць даних), треба утворити єдиний об'єкт, який містив би yci ці зміін1. Це можна зробити, з’єднавши відповідні зм1нні у єдиний вектор, користуючись звичайною операцією створення вектору типу:

x = [xl x2... xN].

Тоді виведення значень кількох змінних у один рядок матиме вигляд:

*disp* ([х1 х2... xN]).

Наведемо приклад:

»xl=1.24; х2=-3.45; хЗ=5.76; х4=-8.07;

» disp([xl x2 хЗ х4])

1.2400 -3.4500 5.7600 -8.0700.

Аналогічно можна об'єднувати кілька текстових змінних, наприклад:

»xl=' psi '; х2=' fi '; хЗ=' teta '; x4=' w1 ';

» disp([xl x2 хЗ x4])

psi fi teta w1

Набагато складніше з'єднувати у єдиний рядок текст i значення змінних, що є також часто потрібно. Труднощі виникають тому, що текстові i числові змінні не можуть з'єднуватися у єдиний вектор. Одним з шляхів подолання цієї перепони є переведення числового значення числової змінної у текстову форму. Це можливо, якщо скористатися функцією *num2str*, яка здійснює таке перетворення. Запис

у = *num2str*(x)

перетворює числове значення змінної «х» на її текстове подання.

Якщо Т - текстова змінна, або деякий текст, а X - числова змінна, то виведення їx у одному рядку можна здійснити зверненням

disp([T num2str(X)]).

Розглянемо приклад:

X =

-9.3088е-015

» Т = 'Значення параметра дорівнює';

» disp([T num2str(x) ])

Значення параметра дорівнює -9.3088е-015

Як випливає з цього прикладу, використання функції num2str – приводить до з'єднання текстової і числової змінних.

Введення інформації з клавіатури у діалоговому режимі можна здійснювати за допомогою функції *input*. Звернення до неї вигляду:

х = *input* ('<запрошення>')

приводить до таких дій ПК. Виконання операторів програми припиняється. ПК переходить до режиму очікування закінчення введення інформації з клавіатури. Після закінчення введення з клавіатури (яке визначається натисканням клавіші <Enter>) введена інформація запам'ятовується у пpoгpaмi під ім'ям "х", i виконання програми продовжується.

Зручним інструментом обрання деякої з альтернатив наступних обчислювальних дій є функція *menu* MatLAB, яка утворює поточне меню користувача. Функція *menu* має таке звернення:

k = menu ( 'Заголовок меню', 'Альтернатива 1, 'Альтернатива 2', ...

, 'Альтернатива n').

Таке звернення приводить до появи на екрані дисплея меню вигляду:

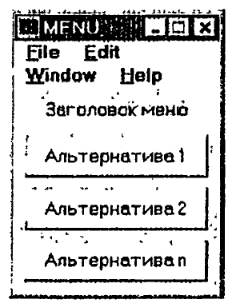


Рис. 1

Виконання програми тимчасово припиняється. Система очікує відповіді у вигляді натискання лiвoї клавіши "мишки" на одну з поданих кнопок меню з альтернативами. Після правильної відповіді вихідному параметру "к" присвоюється значення номера обраної альтернативи (1, 2, …). У загальному випадку кількість альтернатив може бути до 32. Тепер, залежно від одержаного значення цього параметра, можна побудувати процес розгалуження обчислень, наприклад, обрання потрібного для змінювання параметра.

Команда *pause* тимчасово припиняє виконання програми доти, поки користувачем не буде натиснуто будь-яку клавішу клавіатури. Якщо після вказання назви вказати у дужках деяке додатне ціле число *n*, то затримка виконання програми буде здшснюватися протягом *n* секунд.

Якщо у тексті М-файлу зустрічається команда *keyboard*, то при виконанні програми виконання М-файлу припиняється i керування передається клавіатурі дисплея. Цей спеціальний режим роботи позначається появою у командному вікні MatLAB нового виду запрошення до дій

*к* ».

У цьому режимі користувач може здійснити будь-які дії, перевірити або змінити змінні. При цьому йому доступні yci команди i процедури системи MatLAB. Для завершення роботи у цьому режимі необхідно набрати команду *return*. Тоді система продовжить роботу програми з оператора, наступного відносно команди *keyboard*.

1.5. *Типова структура i оформления Script-файлу*

При написанні тексту самостійно виконуваної програми у вигляді Script-файлу слід брати до уваги наступне.

1. Зручно оформлювати весь процес змінювання параметрів у вигляді окремого Script-файлу, наприклад, за ім'ям "ScrFiLMenu", де під скороченням "ScrFil" розуміється ім'я головного Script-файлу.

2. Через те, що при змінюванні параметрів до меню одразу повинні виводитися значення цих параметрів, перед головним циклом програми, який забезпечує повернення до початку обчислень, потрібно розташовувати частину програми, яка б визначала початкові значення ycix параметрів. Окрім того, на початку роботи програми дуже зручним є виведення на екран стислої інформації про призначення програми, більш детальної інформації про досліджувану математичну модель з вказанням місця в ній i змісту ycix початкових параметрів, а також початкових ("вшитих") значень ycix napaмeтpiв цiєї моделі. Це бажано зробити також у вигляді окремого Script-файлу, наприклад, за ім'ям "ScrFil\_Zastavka".

3. Наприкнці програми звичайно виникає потреба дещо упорядкувати робочий npocrip, наприклад, очистити його від введених глобальних змінних (вони, залишаючись у робочому просторі, перешкоджають виконанню іншої наступної програми, яка може мати зовсім інші глобальні змінні, або тaкiж за iм'ям, але інші за значенням), закрити відкриті програмою гpaфичнi вікна (фiгypи) тощо. Цю завершальну частину теж можна оформити як окремий Script-файл, наприклад, назвавши його "ScrFil\_Kin".

У цілому типова схема оформлення самостійного Script-файлу може

бути поданою у наступному вигляді:

% <Позначення Script-файлу (ScrFil.m)>

% <Текст коментаря з описом призначення; програми>

<Порожній рядок>

% Автор < Прізвище І.Б., дата утворення, організація>

%

ScrFil\_Zastavka

k = menu(' Що робити ? ','Продовжити роботу',' Закінчити роботу' );

if k==1,

while k==l

ScrFil\_Menu

ScrFile\_Yadro

k = menu (' Що робити ? ','Продовжити роботу',' Закінчити роботу'');

end

end

ScrFil\_Kin

**Завдання**

Завдання. Створити процедуру функцію обчислення суми нескінченного ряду з заданою точністю ε. Визначити кількість членів, необхідних для досягнення заданої точності ε. Створити програму, що за допомогою створеної підпрограми. виводить до командного вікна таблицю результатів обчислень для різних значень параметру ε=10-2 ;10-3; 10-4 ;10-5 . Варіанти вибрати з таблиці:

| №  вар. | Загальний член ряду | Х |
| --- | --- | --- |
| і |  | 0,12 |
| 2 |  | 0,811 |
| 3 |  | 0,54 |
| 4 |  | 0,21 |
| 5 |  | 0,27 |
| 6 |  | 0,85 |
| 7 |  | 0,29 |
| 8 |  | 0,72 |
| 9 |  | 0,19 |
| і0 |  | 0,64 |
| 11 |  | 0,56 |
| 12 |  | 0,63 |
| 13 |  | 0,4 |
| 14 |  | 1,2 |
| і5 |  | 0,8 |

**Запитання**

1. Які особливості Script- файлів?

2. Яка різниця між Script- файлами і файл-функціями?

3. Які оператори можна використовувати для введення i виведення інформації у діалоговому режимі?

Література

1. Ю.Ф. Лазарев. Початки програмування у середовищі *MatLab* /Навчальний посібник. –К.: Корнійчук, 1999. – 160 с.